On Effect of Communication Delay and Packet Reordering on Estimation and Control

Vijay Gupta

Abstract—Design of communication protocols for real time estimation and control is an important problem in cyberphysical systems. Control performance is dependent on both the throughput and the delay characteristics of the communication network. We consider the problem of estimation and control of a linear time-invariant process across a communication channel that stochastically introduces both a time-varying delay and an erasure event for the sensor data at every time step. Thus, the data may arrive at the receiver delayed and out of order, or may simply be lost. There are two contributions of this work. We prove that even if the delay distribution has an infinite support and the delays are correlated across time, the necessary and sufficient condition for stability depends merely on the erasure probability. We also show that performance of the plant cannot be characterized through a few moments of the delay distribution. Thus, conjectures such as a delay distribution with lesser variance always yields better performance, as compared to a delay distribution with the same mean but larger variance, are incorrect.

I. INTRODUCTION

Cyberphysical systems are expected to be the next generation of engineering systems. Such systems involve significant new features such as feedback control loops closed across multiple scales, networking of numerous dynamic processes to achieve a joint goal, and embedding of processing into physical processes. A systematic design theory for such systems would involve co-design of control, networking, and processing algorithms.

In this paper, we concentrate on a problem in the co-design of control and networking protocols. Traditional network protocols aim at maximizing metrics such as throughput. However, performance of real-time estimation and control depends not only on throughput, but also on the delay introduced by the network. On the other hand, traditional control algorithms are designed by assuming perfect communication among the sensor, the controller, and the actuator. However, communication links introduce many potentially detrimental phenomena, such as quantization error, random delays, data loss and data corruption to name a few, that may lead to performance degradation or even stability loss. Thus, methods to counter the effect of communication protocols on the control design, and to design control oriented networking protocols are needed.

The first of these research directions has received significant attention over the last few years (see, e.g., [1], [4] and the references therein). Two main architectures for design of control loops in the presence of communication networks have emerged:

1) One block design: A compensator block situated at the output of the communication channel (hence co-located with the estimator or the controller) is designed to optimally compensate for the imperfections introduced by the communication channel. Such designs have been considered for communication channels that introduce data loss (see, e.g., [24]) or delay (see, e.g., [15]).

2) Two block design: Both an encoder at the input of the channel and a decoder at the output of the channel are designed. The decoder is collocated with the estimator or the controller. Such design have been considered for digital noiseless channels (e.g., [19]), channels that introduce data loss (e.g., [7]), and channels that introduce additive white Gaussian noise (e.g., [3]).

However, much less work has been done on the design of networking protocols that are suitable for real time control across the network. One reason for this is the fact that the effect of the metrics of network performance, such as throughput, reliability, and delay, on the control performance is not well-understood. Thus, basic questions such as whether a network that supports a higher data rate at the expense of larger delay is more suitable for control than a network that supports a lower data rate, but with a smaller delay, are not yet answered. Consequently, the first step towards extending the design of protocols for control is to characterize the effect of these metrics on the control performance.

In this work, we consider the two block design problem of estimation across communication links that exhibit both data delay and data loss. We aim to obtain the optimal encoder and decoder design, identify fundamental limits on performance, and provide necessary and sufficient conditions for stability in the presence of delay and data loss. We consider the possibility of data reordering due to stochastic delays, and delay distributions with infinite support. We show that the stability conditions are independent of the properties of the delay distribution even if delays are correlated across time. Moreover, we show that the performance cannot be characterized by a few moments of the delay distribution. Thus, naïve folk-theorems that suggest that performance is better with a delay distribution with a smaller mean, and hence a network protocol that minimizes average delay optimizes control performance, are incorrect.

In the control literature, there is significant work in estimation and control across links that introduce either delay or loss.
Concentrating merely on the data loss effect of the channel, both stability [30] and performance [14] have been analyzed. Within the one block design framework, various approaches to compensate for the data loss to counteract the degradation in performance for control costs (see, e.g., [5], [22], [14], [26], [2], [12], [24] and similar works) and estimation costs (e.g., [25], [6]) have been proposed. The two-block design paradigm has also been considered in works like [8], [7], [9] for channels between sensor and the controller, and [13], [16], [18] for channels between controller and the actuator. Similarly, many works have considered channel models that introduce stochastic delay. Using models such as delays being independent and identically distributed from one time step to the next, or delays occurring according to a Markov chain, compensators for delays have been proposed in works such as [22], [15], [10], [27], [30], [28]. Works such as [17], [20], [29] can also be viewed as considering delays in networked control systems. However, most of these works considered delay distributions with finite support, and assumed no packet reordering.

The most closely related works to the present paper are the significant works of Schenato [23] and Robinson and Kumar [21] that consider estimation across channels that introduce both delays and data loss. By assuming that the delay distribution has a finite support (in other words, there is a maximum delay that data suffers, otherwise it is considered to be erased), Schenato [23] showed that stability of the estimate error covariance depends only on the erasure probability. Moreover, a sub-optimal one block design and a two block design were proposed and analyzed with the additional assumption that there is enough memory at the estimator to implement a buffer equal in length to the maximum delay that can be suffered. This assumption was required to counter the effects of packet reordering. Robinson and Kumar [21] analyzed the problem with a long packet assumption that means that the sensor could transmit all the previous measurements at every time step (in other words, the sensor transmits a vector of unbounded dimension at every time). With this assumption, they showed that if the delays are independent and identically distributed, then they do not impact the stability of the estimation error covariance. We provide a design that involves transmitting a finite dimensional vector at every point, and also consider the effect of correlated delays. We also characterize the effect of delays on performance.

In this work, we consider delay distributions that can have infinite support, as well as packet erasure. We permit the possibility of packet reordering. We begin by presenting the solution to the two block design problem. We prove that the delay distribution does not impact stability, even if the delays are correlated from one time step to the next. We then characterize the effect of delay on the performance and present some counter-examples to the intuitive result that the performance is necessarily improved by lowering the mean or the variance of the maximum of the delay distributions. This has potential design implications in designs of control oriented network protocols.

II. FRAMEWORK DESCRIPTION AND PROBLEM FORMULATION

Consider a process evolving as

\[ x(k+1) = Ax(k) + w(k), \quad k \geq 0 \]  

where \( x(k) \in \mathbb{R}^n \) is the process state and \( w(k) \) is the process noise assumed to be white, Gaussian, zero mean with covariance \( R_w > 0 \). The initial state \( x(0) \) is a zero mean and Gaussian random variable with covariance matrix \( P(0) \). The process state is observed using a sensor that generates measurements, or observations, of the form

\[ y(k) = Cx(k) + v(k), \quad k \geq 0 \]

where \( y(k) \in \mathbb{R}^m \) and the measurement noise \( v(k) \) is also assumed to be white, Gaussian, zero mean with positive definite covariance matrix \( \Sigma_v \). We assume that the pair \( (A, C) \) is observable.

The sensor communicates with an estimator across a communication channel. In this paper, we are mostly interested in the two block design problem, in which the designer specifies an encoder block situated at the input of the channel (collocated with the sensor) and a decoder block at the output of the channel (collocated with the estimator). In Section III-F, we briefly consider the implications of our results to the single block design case in which the sensor transmits measurements \( y(k) \) over the channel and only the decoder block at the output of the channel needs to be designed. In the two block design problem, at every time step \( k \), the sensor transmits information in the form of a finite-dimensional vector \( s(k) \) across the channel to the estimator. The communication channel introduces stochastic time-varying delay with packet reordering and erasure (equivalent to infinite packet delay) being possible. We assume the provision of a time-stamp so that the estimator knows the time at which the sensor transmitted any vector \( s(k) \). In general, the delay suffered by various transmitted packets can be correlated, or dependent on when the transmission occurred. While our results can be generalized to these cases as discussed in Section III-D, for pedagogical ease, we will usually assume the delays to be independent and identically distributed (i.i.d.) from one time step to the next.

For an i.i.d. distribution of delay, for any value of the delay \( m \geq 0 \), denote the cumulative distribution function of the delay suffered by any transmitted packet being at most \( m \) steps by \( F(m) \). Thus, the probability \( p(m) \) of delay being equal to \( m \) time steps is given by \( F(m) - F(m-1) \). Moreover, the probability of erasure is given by \( F(m) \) as \( m \rightarrow \infty \). Note that both packet reordering as well as the possibility of the decoder receiving multiple packets at the same time step are allowed at any time step. Denote the event of the transmitted data suffering a delay of \( m \) time steps by \( d(k) = m \), with \( m \rightarrow \infty \) denoting packet erasure. A packet that is received is assumed to be without any errors. We will revisit the assumption of the delays \( d(k) \) being independent and identically distributed in Section III-D. Finally, we assume that the sources of...
randomness $x(0)$, $\{d(k)\}_{k=0}^\infty$, $\{v(k)\}_{k=0}^\infty$, and $\{w(k)\}_{k=0}^\infty$ are mutually independent.

The information $s(k)$ transmitted by the encoder or the sensor at time $k$ is a finite $d$-dimensional real vector that can depend on the measurements $\{y(t)\}_{t=0}^k$. Note that we do not assume that the encoder knows the delays suffered by the previous packets through some mechanism such as an acknowledgement. The estimator has access to the outputs of the channel till time $k$, i.e. the set $\{s(t) : 0 \leq t \leq k, d(t) + t \leq k\}$. Using these outputs, it generates an estimate $\hat{x}_{dec}(k+1)$ of the process state $x(k+1)$ to minimize the estimation error covariance

$$P(k+1) = E[(x(k+1) - \hat{x}_{dec}(k+1))(x(k+1) - \hat{x}_{dec}(k+1))^T],$$

where the expectation is taken over the initial condition, and the process and measurement noises. Thus, the estimator is a minimum mean squared error estimator. Due to the stochastic delays and erasures introduced by the channel, $P(k)$ is a random variable. Thus, we need to characterize various moments of $P(k)$. We will concentrate on the expected value of $P(k)$, although the techniques we use can easily be extended to obtain arbitrary moments. In particular, if the term $\lim_{k \to \infty} E[P(k)]$ is bounded, we say that the estimate error is stable.

Given the above assumptions, we wish to analyze the stability and performance of the system as a function of the probability mass function of the delay for the optimal design of the encoder (the vector $s(k)$) and the estimator (the estimate $\hat{x}_{dec}(k)$). In particular, we consider the following problems:

1) What are the conditions on the delay distribution and erasure probability so that the estimation error cannot be stabilized for any encoder-decoder design?
2) If an encoder and decoder can stabilize the error, what is the minimum error covariance that is achievable?
3) How does the error covariance depend on the delay distribution? Do the first few moments of the distribution (e.g. the mean and the variance) suffice to determine the error covariance?

Relation to the Control Problem: The estimation problem stated above is closely related to the control of a linear time-invariant process across a channel that introduces stochastic delays and erasures. Consider, thus, a process evolving as

$$x(k+1) = Ax(k) + Bu(k) + w(k) \quad k \geq 0$$

where $u(k) \in \mathbb{R}^d$ is an additional control variable that needs to be designed to minimize the standard quadratic cost

$$J_K = E\left[\sum_{k=0}^{K} \left(x^T(k)Qx(k) + u^T(k)Ru(k)\right)\right.$$

$$\left. + x^T(K+1)P(K+1)x(K+1)\right],$$

for positive definite matrices $Q$, $R$, and $P(K+1)$. This problem formulation of a linear process, Gaussian noises, and a quadratic cost is the so-called linear quadratic Gaussian (LQG) problem setup, that is a classical control formulation. The non-classical element here is the presence of a communication channel. The assumptions about the initial state $x(0)$, the sensor, the process and measurement noises, the encoder and the delays introduced by the channel are identical to the ones stated above. The estimator at the output of the channel is replaced by a controller that calculates and transmits a control input $u(k)$ at every time step, that is applied to the process in (3). We assume that the pair $(A, C)$ is observable, and the pair $(A, B)$ is controllable.

It is fairly straight-forward to prove a separation principle for the control problem along the lines of [11, Chapter 10] (see also [9, Proposition 1]). The principle states that the optimal control input $u(k)$ for the problem is calculated by assuming that the communication channel and the noises were absent, and the controller had access to the state $x(k)$ at every time step (the so-called linear quadratic regulator (LQR) optimal control law). The only variation is that the controller uses the minimum mean squared error estimate of the state $x(k)$ instead of the state $x(k)$ itself. Thus, the separation principle states that the optimal control input for our problem can be calculated in two steps:

1) Calculate the minimum mean squared error estimate of $x(k)$.
2) Use this estimate as the correct value of the state in the LQR optimal control law.

In general, the estimate $\hat{x}(k)$ may be a function of both the measurements $y(0), y(1), \ldots, y(k-1)$, and the previous control inputs $u(0), \ldots, u(k-1)$. However, since all the previous control inputs are known to the controller, the effect of the control inputs can be canceled and only the estimate of the state based on the previous measurements needs to be calculated. Thus, the LQG control problem reduces to the estimation problem posed earlier. Stability conditions for the control and estimation problems are also identical. Thus, from now on, we will focus on the estimation problem while keeping in mind that the solution of the control problem is a simple extension as stated above.

### III. Main Results

At every time step $k$, define a set $D(k)$ as

$$D(k) = \{ j : j + d(j) \leq k \} \cup \{-1\}.$$ 

Thus the set $D(k)$ consists of time steps such that the vectors $s(j)$ transmitted at those time steps have been received by the decoder, and the element -1. Define the time stamp $t_s(k)$ as the maximal element in the set $D(k)$:

$$t_s(k) = \max_j \text{ such that } j \in D(k).$$

Thus the time stamp $t_s(k)$ defines the last time such that the vector $s(t_s(k))$ has been received at the decoder. Moreover, if no packet has been received, $t_s(k) = -1$. 

A. Optimal Encoder and Decoder

Denote the estimate of the state \( x(k) \) given the measurements \( \{y(j)\}_{j=0}^{t} \) by \( \hat{x}(k|y(0), \ldots, y(t)) \). Consider the following encoder and decoder design that is an extension of the design proposed in [7] in the context of channels that introduce erasures but no stochastic delays:

1. **Encoder Design:** The encoder calculates an estimate \( \hat{x}(k|y(0), \ldots, y(k)) \) of the state given all measurements till the present time step using a Kalman filter. It transmits the estimate at every time step, along with the time stamp \( k \).

2. **Decoder Design:** The decoder calculates the estimate \( \hat{x}_{\text{dec}}(k+1) \) of the state \( x(k+1) \) at every time step \( k \) as follows. The initial value is \( \hat{x}_{\text{dec}}(0) = 0 \). At time \( k \), there are three possibilities:
   - The decoder does not receive any packet from the encoder. It updates the estimate \( \hat{x}_{\text{dec}}(k+1) = A \hat{x}_{\text{dec}}(k) \).
   - Of the (possibly multiple) packets received by the decoder, the packet with the maximal time-stamp has time stamp \( m \), where \( m \) is larger than the time stamp of any packet that the decoder has received till time \( k-1 \). It sets
     \[
     \hat{x}_{\text{dec}}(k+1) = A^{k-m} \hat{x}(m+1|y(0), \ldots, y(m)),
     \]
     using the packet with time stamp \( m \).
   - Of the (possibly multiple) packets received by the decoder, the packet with the maximal time-stamp has time stamp \( m \), where \( m \) is larger than the time stamp of at least one packet that the decoder has received till time \( k-1 \). It ignores the packets and updates
     \[
     \hat{x}_{\text{dec}}(k+1) = A \hat{x}_{\text{dec}}(k).
     \]

**Proposition 3.1: (Optimality of the Encoder-Decoder Design.)** Consider the problem formulation as stated in Section II. The encoder-decoder design given above leads to the estimate with the minimum error covariance at the decoder among all causal designs, at every time step.

**Proof:** The proof follows from two observations:

1. The encoder-decoder design given above leads to the estimate \( \hat{x}_{\text{dec}}(k+1) = \hat{x}(k+1|t_s(k)) \) at every time \( k \). In other words, the estimate at the decoder is identical to the one that would be formed if the decoder had access to all the measurements in the set \( D(k) \).

2. No causal encoder-decoder design can lead to an estimate with a better error covariance than an estimate based on all measurements in the set \( D(k) \). In other words, even if the encoder were to transmit at every time step \( k \) all the information it has access to till time \( k \), the error covariance cannot be lesser than the estimate formed due to the encoder-decoder design proposed above.

**Remarks:**

1. Note that the encoder-decoder design is optimal even though a vector with a constant dimension is recursively calculated and transmitted at every time step. A design that transmitted all measurements till time \( k \) would, on the other hand, involve increasing amount of data transmission.

2. **Optimality for any delay realization and at every time step:** The design proposed above provides the optimal estimate for an arbitrary realization of the delay process, irrespective of whether the delays are i.i.d. or correlated across time. The design also results in the optimal estimate at every time step for any realization of the delay process.

B. Analysis

We begin by characterizing the error covariance \( P(k+1) \) of the estimate \( \hat{x}_{\text{dec}}(k+1) \). Define by \( M(k+1) \) the error covariance of the estimate \( \hat{x}(k+1|y(0), \ldots, y(k)) \). Since the pair \((A, C)\) is observable, \( M(k) \) converges geometrically to a steady state value \( M^* \). Also define by \( f(S) \) the Lyapunov recursion

\[
f(S) = ASA^T + R_w.
\]

Thus, \( f(S) \) is the error covariance of the estimate of state \( x(k+1) \), if the error covariance of the estimate of the state \( x(k) \) was \( S \), and no further measurement was received. Finally define

\[
f_m(S) = f(f(\cdots f(S)\cdots)),
\]

applied \( m \) times with the convention that \( f_0(S) = S \).

The expected error covariance for estimate of state \( x(k+1) \) can be calculated by conditioning it on the values of the time stamp. Thus,

\[
E[P(k+1)] = E[E[(x(k+1) - \hat{x}_{\text{dec}}(k+1))(x(k+1) - \hat{x}_{\text{dec}}(k+1))^T | t_s(k) = m]]
\]

\[
= \sum_{m=-1}^{k} \text{Prob}(t_s(k) = m) E[(x(k+1) - \hat{x}_{\text{dec}}(k+1))(x(k+1) - \hat{x}_{\text{dec}}(k+1))^T | t_s(k) = m],
\]

where the expectation is now only over the initial condition and the noises. Since \( \hat{x}_{\text{dec}}(k+1) = \hat{x}(k+1|t_s(k)) \), we can evaluate

\[
E[(x(k+1) - \hat{x}_{\text{dec}}(k+1))(x(k+1) - \hat{x}_{\text{dec}}(k+1))^T | t_s(k) = m] = f_{k-m}(M(m+1)).
\]

Moreover,

\[
\text{Prob}(t_s(k) = m) = \prod_{j=0}^{k-m-1} \text{Prob}(d(k-j) > j)\text{Prob}(d(m) \leq k - m),
\]

since the packet transmitted at time \( m \) needs to be delayed by no more than \( k - m \) time steps, and all packets after that...
time should suffer enough delay not to be received till time \( k \). Since the delays are i.i.d., we can rewrite
\[
\text{Prob}(t_s(k) = m) = \prod_{j=0}^{k-m-1} \text{Prob}(d(k) > j)\text{Prob}(d(k) \leq k - m).
\]
This expression holds for values of \( m \geq 0 \). For \( m = -1 \),
\[
\text{Prob}(t_s(k) = -1) = \prod_{j=0}^k \text{Prob}(d(k) > j).
\]
Thus, the expected error covariance evaluates to
\[
E[P(k+1)] = \sum_{m=0}^k \left( \prod_{j=0}^{k-m-1} \text{Prob}(d(k) > j)\text{Prob}(d(k) \leq k - m) \right) f_{k-m}(M(m+1)) + \prod_{j=0}^k \text{Prob}(d(k) > j)f_{k+1}(P(0)). \tag{5}
\]

**Remark:** Note that the above technique can be used to obtain arbitrary moments of \( P(k+1) \) at any time step. Thus, the entire probability mass function of the error covariance can be characterized.

### C. Stability

We can now characterize the stability conditions for the system. Because of Proposition 3.1, these stability conditions are necessary for stability with any encoder and decoder design. For the optimal encoder and decoder design presented in the previous section, these conditions are both necessary and sufficient.

**Proposition 3.2 (Stability Conditions):** Consider the problem formulation as stated in Section II. A necessary condition for stability with any causal encoder decoder design is given by the inequality
\[
\lim_{m \to \infty} (1 - F(m))\rho(A)^2 < 1. \tag{6}
\]
Conversely, the encoder decoder design specified in Section III-A leads to stability if (6) is satisfied.

**Proof:** The proof follows from error covariance given in (5). To prove necessity, we show that a lower bound for the expected error covariance diverges unless (6) is satisfied.

From (5), we obtain
\[
E[P(k+1)] \geq \sum_{m=0}^k \left( \prod_{j=0}^{k-m-1} \text{Prob}(d(k) > j) \right) \text{Prob}(d(k) \leq k - m) f_{k-m}(M(m+1)) \\
\geq \prod_{j=0}^{k-m-1} \text{Prob}(d(k) > j) \text{Prob}(d(k) \leq k - m) f_{k-m}(M(m+1)) \\
= \prod_{j=0}^{k-m-1} \text{Prob}(d(k) > j) \text{Prob}(d(k) \leq k - m) f_{k-m}(M(m+1)),
\]
where the first inequality follows since \( f_{k+1}(P(0)) \) is positive semi-definite, and the second inequality follows since \( \text{Prob}(d(k) \leq k - m) \geq \text{Prob}(d(k) = 0) \) for any \( 0 \leq m \leq k \). Now, for stability, we are interested only in large values of \( k \).

Since \( M(k) \to M^* \) geometrically, we can substitute \( M(m+1) \) by \( M^* \) in the above expression. Thus, we need to study the convergence of the lower bound given by
\[
\Lambda(k+1) = \text{Prob}(d(k) = 0) \sum_{m=0}^k f_{k-m}(M^*) \\
\prod_{j=0}^{k-m-1} \text{Prob}(d(k) > j) \\
\geq \prod_{j=0}^{k-m-1} \text{Prob}(d(k) > j) \text{Prob}(d(k) \leq k - m) f_{k-m}(M(m+1)) \\
= \prod_{j=0}^{k-m-1} \text{Prob}(d(k) > j) \text{Prob}(d(k) \leq k - m) f_{k-m}(M(m+1)),
\]
where the inequality follows from the definition of the Lyapunov recursion \( f(\cdot) \) and the fact that \( R_w \) is positive definite, and we define
\[
T_m = A^{k-m}(M^*)(A^{k-m})^T \prod_{j=0}^{k-m-1} (1 - F(j)).
\]
A necessary condition for convergence of \( \Lambda(k+1) \) is that the sum of terms \( T_m \) converges, where \( T_j \) is related to \( T_{j-1} \) through the discrete Lyapunov recursion
\[
T_j = (1 - F(k - j))AT_{j-1}A^T. \tag{7}
\]
Since all terms $T_j$'s are positive semi-definite, a necessary condition for convergence of the sum is that each term $T_j$ converges. For stability we consider $k \to \infty$. Thus, a necessary condition for the individual terms $T_j$ to converge through the Lyapunov recursion (7) is that

$$\lim_{m \to \infty} (1 - F(m)) \rho(A)^2 < 1.$$ 

Thus a necessary condition for stability is indeed given by (6). To prove sufficiency, we consider an upper bound for $E[P(k + 1)]$. We have

$$E[P(k + 1)] \leq \sum_{m=0}^{k} f_{m}(M(m + 1))$$

which is another discrete Lyapunov recursion that converges as $k \to \infty$ if (6) holds. Thus, the stability condition in equation (6) is sufficient for stability.

The above result shows that only the erasure probability can impact stability. That a delay distribution that has finite support and does not induce packet reordering cannot impact stability was shown in [23] in the context of single block design. This result shows that even delay distributions with infinite support and packet reordering do not alter stability conditions. The stability condition in equation (6) is identical to the one that was shown in [8] to be necessary and sufficient for the case when the channel introduces only packet erasure, since the erasure probability is given by $F(m)$ as $m \to \infty$.

**D. Correlated delays**

While the discussion above has focussed on delays that are independently and identically distributed from one time step to the next, the analysis can be generalized to consider correlated delays. If the delays are correlated, we can still follow the derivation in Section III-B to obtain the expected value of the error covariance. The expression for time stamp in equation (4) in this case is given by

$$\text{Prob}(t_s(k) = m) = \text{Prob}(d(k) > 0, d(k - 1) > 1, \ldots, d(m + 1) > k - m - 1, d(m) \leq k - m).$$

Expected error covariance expressions along the lines of (5) can now be derived. If the delay distribution reaches a steady state, simple stability conditions can also be derived. For example, let the delay distribution be given by a Markov chain, such that $q_{mn}$ represents the probability that the delay suffered by packet at time $k$ be more than $n$, given that the probability that the delay suffered by packet at time $k - 1$ is greater than $m$. Then, a necessary and sufficient condition for stability is that

$$\lim_{m \to \infty} q_{mm} \rho(A)^2 < 1.$$ 

**E. Performance**

The expression in (5) provides the exact expression for the expected error covariance at any time $k$. Moreover, the technique can be used to characterize any moment of $P(k)$. The expression can be used to constrain the delay guarantees that a communication network must provide for an acceptable level of error covariance performance. However, in practice, it may be desirable to obtain a proxy of the delay through, e.g., its first few moments. Thus, e.g., it may be supposed that a reasonable goal of a network protocol for optimizing the control performance is to minimize the expected delay induced. Similarly, it seems a reasonable conjecture that for two probability mass functions of the delay that have the same means, the function that has a lower variance (i.e., is more tightly constrained) must provide a better performance. In this section, we provide some counter-examples to show that such statements do not hold universally.

Consider a scalar process evolving as

$$x(k + 1) = 1.2x(k) + w(k),$$

where $w(k)$ is a white noise process with variance $\sigma^2$. The mean of $x(k)$ is 0.5, and its variance is 0.4, which is significantly smaller than the variance of $x(k)$ under the second delay distribution.

Consider a second scalar process evolving as

$$x(k + 1) = 1.2x(k) + w(k),$$

where $w(k)$ is a white noise process with variance $\sigma^2$. The mean of $x(k)$ is 0.5, and its variance is 0.4, which is significantly smaller than the variance of $x(k)$ under the second delay distribution.
with noise covariance $R_w = 1$ and the covariance of the initial state $P(0) = 1$. The process is observed by a sensor of the form
\[ y(k) = x(k) + v(k), \]
with noise covariance $R_v = 1$. If all the measurements were available to the estimator without delay, the steady state error covariance $M^* = 1.9522$. Consider two delay probability mass functions:
\[
p_1(m) : \text{Prob}(d = 5) = 1
\]
\[
p_2(m) : \text{Prob}(d = m) = \begin{cases} \frac{1}{5} & m = 0 \\ \frac{4}{5} & m = 6. \end{cases}
\]
Both the probability mass functions $p_1(m)$ and $p_2(m)$ have mean $= 5$. However, $p_1(m)$ has a smaller variance and a smaller maximum value for delay. The steady state expected error covariance for the two functions is given by $E[P(k)] = 23.88$ for $p_1(m)$ and $E[P(k)] = 17.34$ for $p_2(m)$. Thus the probability mass function with the higher variance (and the higher maximum value) has the lower cost. If the process evolves as
\[ x(k + 1) = 2x(k) + w(k), \]
the probability mass function with the higher variance (i.e. $p_2(m)$) has the higher cost. Thus, there is no general relation between the variance or the maximum value of the delay and the cost achieved even if the means of two probability mass functions for the delays are the same.

Similarly, it is not necessarily true that a probability mass function of the delay with a higher mean leads to a higher expected error covariance. For instance, consider the same process as in (9), but with the probability mass functions
\[
p_1(m) : \text{Prob}(d = 2) = 1
\]
\[
p_2(m) : \text{Prob}(d = m) = \begin{cases} \frac{1}{4} & m = 0 \\ \frac{3}{4} & m = 1 \\ \frac{1}{3} & m = 6. \end{cases}
\]
The probability mass functions $p_1(m)$ and $p_2(m)$ have means $2$ and $2.33$ respectively. The steady state expected error covariance for the two functions is given by $E[P(k)] = 6.49$ for $p_1(m)$ and $E[P(k)] = 4.31$ for $p_2(m)$. Thus, lesser mean of the delay probability mass function does not necessarily lead to better performance.

An intuitive reason a finite number of moments do not directly specify the control performance can be obtained by taking a closer look at (5). We can define an auxiliary probability mass function $q(d(k))$ as
\[
q(d(k) = -1) = \prod_{j=0}^{k} \text{Prob}(d(k) > j)
\]
\[
q(d(k) = m) = \text{Prob}(d(k) \leq k - m) \prod_{j=0}^{k-m-1} \text{Prob}(d(k) > j)
\]
for $m = 0, \ldots, k$. It can be verified that this is a valid probability mass function. Using this auxiliary function, the performance can be rewritten as
\[
E[P(k+1)] = \sum_{m=-1}^{k} q(d(k) = m) f_{k-m}(M(m+1)), \tag{10}
\]
with the notation $M(0) = P(0)$. In other words, the expected error covariance is given by the expected value of $f_{k-m}(M(m+1))$. However the expectation is taken not with respect to the delay probability mass function $p(m)$, but with respect to the auxiliary probability mass function $q(d(k))$.

Thus, while it is true that for two probability mass functions $q(d(k))$ with the same mean, a higher variance implies a higher error covariance, a similar statement is not true with $p(m)$.

F. Implications for Single Block Design

While most of our discussion so far has concentrated on the two block design problem, the fact that single block designs are a subset of two block designs implies that there are important implications for the single block designs as well. The optimal two block design proposed above provides a lower bound for the cost achievable using any single block design. In particular, the stability conditions identified in Theorem 3.2 provide necessary stability conditions if the sensor transmits measurements, and the estimator is designed to optimally utilize the received data. Similarly, the expression in equation (5) provides a lower bound on the expected error covariance for this case.

IV. CONCLUSIONS AND FUTURE DIRECTIONS

Control performance is dependent on both the throughput and the delay characteristics of the communication network. We consider characterization of the effect of communication protocols for real time estimation and control. In particular, we study the problem of estimation and control of a linear time-invariant process across a communication channel that introduces both a stochastic delay and erasure for the sensor data at every time step. The delay is time-varying and can cause re-ordering of the transmitted data packets. There are two main contributions of this work. We prove that even if the delay distribution has an infinite support and the delays are correlated across time, the necessary and sufficient condition for stability depends merely on the erasure probability. We also show that performance of the plant cannot be characterized through a few moments of the delay distribution. Thus, conjectures such as the control performance is monotonic with the mean or the variance or the maximum value of the delay distribution are incorrect.

The work is only a first step towards systematic co-design of communication and control protocols. The characterization of the effect of delay on performance is not yet fully satisfactory. It will be useful to obtain rules of thumb (rather than the analytical expressions obtained in the paper) to compare the effect of delays and erasure rate on performance, so that communication protocols can be compared easily. Delays in
communication protocols are often caused by mechanisms such as acknowledgements and retransmissions. For real-time control, it is not clear if such mechanisms are useful. Finally, we would like to move away from analysis and towards a design theory for communication protocols that are control oriented.

ACKNOWLEDGMENTS

Research supported in part by NSF awards 0846631 and 0834771.

REFERENCES


[6] G. N. Nair, F. Fagnani, S. Zampieri and R. J. Evans, “Feedback control oriented. Design theory for communication protocols that are control we would like to move away from analysis and towards a control, it is not clear if such mechanisms are useful. Finally, such as acknowledgements and retransmissions. For real-time communication protocols are often caused by mechanisms such as

ACKNOWLEDGMENTS

Research supported in part by NSF awards 0846631 and 0834771.

REFERENCES


[6] G. N. Nair, F. Fagnani, S. Zampieri and R. J. Evans, “Feedback control oriented. Design theory for communication protocols that are control we would like to move away from analysis and towards a control, it is not clear if such mechanisms are useful. Finally, such as acknowledgements and retransmissions. For real-time communication protocols are often caused by mechanisms such as

ACKNOWLEDGMENTS

Research supported in part by NSF awards 0846631 and 0834771.

REFERENCES


[6] G. N. Nair, F. Fagnani, S. Zampieri and R. J. Evans, “Feedback control oriented. Design theory for communication protocols that are control we would like to move away from analysis and towards a control, it is not clear if such mechanisms are useful. Finally, such as acknowledgements and retransmissions. For real-time communication protocols are often caused by mechanisms such as

ACKNOWLEDGMENTS

Research supported in part by NSF awards 0846631 and 0834771.

REFERENCES


